



**Weiterentwicklung der Stundenplan-App der Hochschule Hof**  
**Integration von Push-Notifications**

**P r o j e k t a r b e i t**

**an der Hochschule für Angewandte Wissenschaften Hof**  
**Fakultät Informatik**  
**Studiengang Mobile Computing**

**Vorgelegt bei**  
**Prof. Dr. Michael Stepping**  
**Alfons-Goppel-Platz 1**  
**95028 Hof**

**vorgelegt von**  
**Jonas Beetz**  
**Steinweg 14**  
**96358 Teuschnitz**

**Hof, 04.07.2017**

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis.....</b>	<b>II</b>
<b>Abbildungsverzeichnis .....</b>	<b>III</b>
<b>Listingverzeichnis .....</b>	<b>III</b>
<b>Anhangsverzeichnis.....</b>	<b>III</b>
<b>Abkürzungsverzeichnis.....</b>	<b>IV</b>
<b>1    Einleitung .....</b>	<b>1</b>
1.1    Funktionsweise .....	2
1.2    Anlegen der Datenbanken .....	2
1.3    Umsetzung.....	4
<b>2    Fazit &amp; Ausblick.....</b>	<b>5</b>
<b>Anhang .....</b>	<b>6</b>

## **Abbildungsverzeichnis**

Abbildung 1: Kommunikation zwischen den Servern und der App .....	2
---	---

## **Listingverzeichnis**

Listing 1: Tabellenstruktur für die Tabelle `fcm_nutzer` .....	3
Listing 2: Indizes für die Tabelle `fcm_nutzer` .....	3
Listing 3: Tabellenstruktur für die Tabelle 'fcm_verlegungen'.....	3
Listing 4: Indizes für die Tabelle 'fcm_verlegungen'.....	3

## **Anhangsverzeichnis**

Anhang 1: fcm_connect_db.php -Datei .....	6
Anhang 2: fcm_register_user.php -Datei Teil 1 .....	7
Anhang 3: fcm_register_user.php -Datei Teil 2 .....	8
Anhang 4: fcm_update_and_send.php -Datei Teil 1 .....	9
Anhang 5: fcm_update_and_send.php -Datei Teil 2 .....	10
Anhang 6: fcm_update_and_send.php -Datei Teil 3 .....	11

## **Abkürzungsverzeichnis**

ID	Identifikator
PHP	Hypertext Preprocessor
SQL	Structured Query Language

# **1 Einleitung**

Im Rahmen dieser Arbeit wurde das Thema Push-Notifications für die Android Stundenplan-App der Hochschule Hof aufgegriffen. Für die Umsetzung wurde der von Google bereitgestellte Dienst Firebase Cloud Messaging (folgend Firebase abgekürzt) verwendet. Firebase stellt unter anderem Funktionalitäten zum Verwalten von Tokens bereit. Jeder Token steht dabei stellvertretend für ein Endgerät, dass sich für eine Push-Nachricht registriert hat.

Die Umsetzung erfolgte mit Hilfe eines von der Hochschule Hof bereitgestellten App-Servers. Der Server kann für Datenbankverwaltung und Bereitstellen von Schnittstellen genutzt werden. Im Zuge der Programmierarbeiten wurden mehrere PHP-Skripte erstellt. Diese können ebenso auf dem Server abgelegt und anschließend von diesem zur Verfügung gestellt werden. Die Skripte sind für das Ändern der Datenbank, die Kommunikation mit dem Firebase Server und Kommunikation zwischen Hochschul-Server und App notwendig.

## 1.1 Funktionsweise

Dieser Abschnitt erklärt die Funktionsweise, wie Push-Notifications an die App versandt werden. Abbildung 1 zeigt die Kommunikation zwischen dem Firebase Server, der Server & Datenbank der Hochschule Hof und der Stundenplan-App für Android der Hochschule Hof.

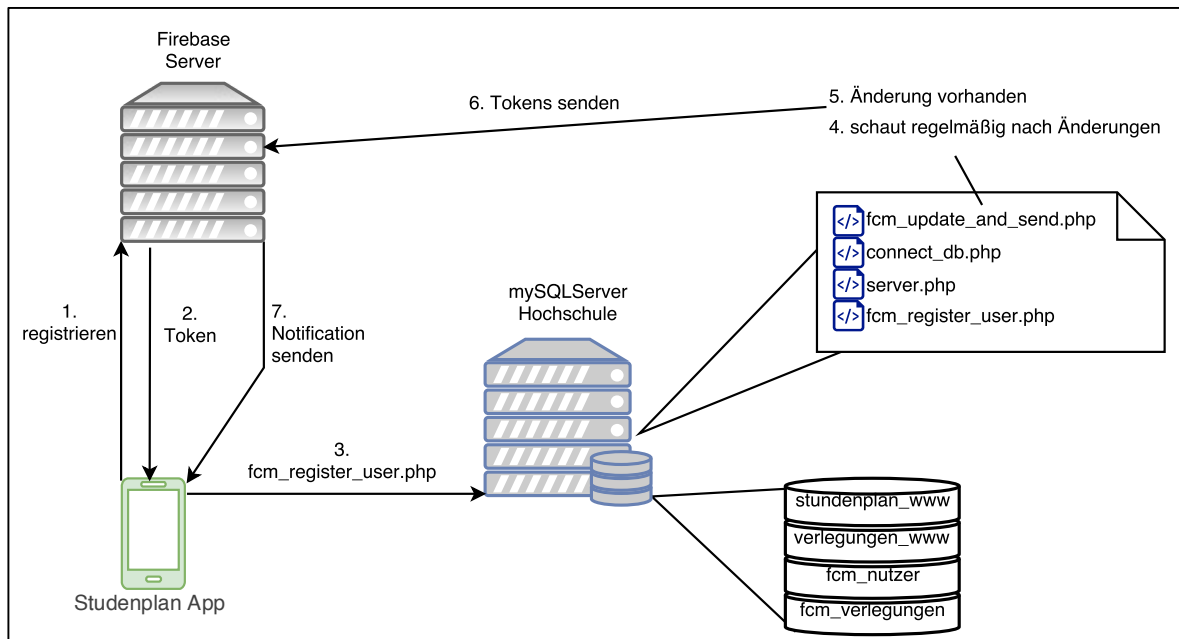


Abbildung 1: Kommunikation zwischen den Servern und der App

1. Jedes Gerät registriert sich beim ersten Starten der App am Google Server (1) und bekommt (2) einen eindeutigen Token zugewiesen. Das Gerät sendet einen Token zusammen mit allen Vorlesungs-IDs der Vorlesungen, welche der Nutzer besucht, an den Hochschulserver. Dort wird die PHP Datei `fcm_register_user.php` aufgerufen. Diese speichert den Token und alle zugehörigen Vorlesungs-IDs in der Tabelle `fcm_nutzer` (3). Das PHP Script `fcm_update_and_send.php` wird in regelmäßigen Abständen aufgerufen (4). Dieses schaut nach neuen Änderungen in der `verlegungen_www` Tabelle. Sobald eine neue Änderung vorhanden ist, wird der Firebase Server benachrichtigt. Hierbei werden alle Tokens an den Firebase Server übergeben, an welche dieser eine Notification senden soll (5-6). Der Firebase Server sendet eine Notification an alle übergebenen Tokens (7).

## 1.2 Anlegen der Datenbanken

Für die Integration von Push Notifications in die Stundenplan App für Android der Hochschule Hof ist es nötig eigene Tabellen für diesen Dienst anzulegen. Diese werden am

MySQL-Server der Hochschule Hof angelegt. Die Tabelle *fcm\_nutzer* verwaltet die Tokens aller registrierten Geräte und all deren zugehörige Vorlesungen.

Listing 1 zeigt den SQL-Befehl für das Anlegen der Tabelle *fcm\_nutzer*.

```
CREATE TABLE `fcm_nutzer` (  
  `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `token` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `vorlesung_id` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL
```

Listing 1: Tabellenstruktur für die Tabelle *fcm\_nutzer*

Um einen eindeutigen Zugriff auf einzelne Spalten der Tabelle gewährleisten zu können ist es notwendig einen Index für die Spalten *token* und *vorlesung\_id* zu erstellen. Den hierfür notwendigen SQL-Befehl bildet Listing 2 ab.

```
ALTER TABLE `fcm_nutzer`  
  ADD INDEX `token` (`token`),  
  ADD INDEX `vorlesung_id` (`vorlesung_id`):
```

Listing 2: Indizes für die Tabelle *fcm\_nutzer*

Die Tabelle *fcm\_verlegungen* beinhaltet alle IDs von Stundenplanänderungen und die dazugehörigen Vorlesungen. Das Anlegen dieser Tabelle wird in Listing 3 dargestellt.

```
CREATE TABLE `fcm_verlegungen` (  
  `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `verlegung_id` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL,  
  `vorlesung_id` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL
```

Listing 3: Tabellenstruktur für die Tabelle *fcm\_verlegungen*

Auch für die Spalten der Tabelle *fcm\_verlegungen* ist es wichtig diese mit einem Index zu versehen. Der SQL-Befehl für das Erstellen des Indizes wird in Listing 4 dargestellt.

```
ALTER TABLE `fcm_verlegungen`  
  ADD INDEX `verlegung_id` (`verlegung_id`),  
  ADD INDEX `vorlesung_id` (`vorlesung_id`):
```

Listing 4: Indizes für die Tabelle *fcm\_verlegungen*



### **1.3 Umsetzung**

Google stellt für die Verwendung von Firebase Cloud Messaging in Android Apps eigene Methoden zur Verfügung. Diese müssen von der App richtig interpretiert und genutzt werden. Sie sind unter anderem dafür zuständig, dass die App sich am Firebase Server mit einem Token registrieren, oder eingehende Nachrichten vom Server empfangen kann. Des Weiteren wurden PHP Skripte entwickelt, die sämtliche Aufgaben zur Kommunikation zwischen den Servern, außerhalb der App, übernehmen. Diese sind in den Anhängen 1 bis 6 beigefügt.

## **2     Fazit & Ausblick**

Es wurde eine vollständig funktionsfähige Erweiterung zum Empfangen von Push-Notifications in der Android Stundenplan App der Hochschule Hof entwickelt und umgesetzt. Diese läuft derzeit auf einer Testumgebung, welche sich jedoch kaum von der Live-Umgebung unterscheidet. Es bedarf daher nur noch wenig Aufwand an Weiterentwicklung, um das Projekt künftig im Liveeinsatz nutzen zu können.

# Anhang

```
1  <?php
2  require_once 'passwords.php';
3
4  $host = "localhost";
5  $db_name = "t3_ext";
6  $db_user = "fcmuser";
7  $db_password = $fcmuserpassword;
8
9  $con = new mysqli($host, $db_user, $db_password, $db_name);
10
11 /* check connection */
12 if (mysqli_connect_errno()) {
13     printf("Connect failed: %s\n", mysqli_connect_error());
14     exit();
15 }
16
17 /* change character set to utf8 */
18 if (!$con->set_charset("utf8")) {
19     printf("Error loading character set utf8: %s\n", $mysqli->error);
20     exit();
21 }
22 ?>
```

Anhang 1: fcm\_connect\_db.php -Datei

```

1 <?php
2
3 /*
4
5 Testlinks:
6
7 https://app.hof-university.de/soap/fcm_register_user.php?debug=1
8 muss allerdings mit einem Tool wie z.B. Advanced Rest Client
9 https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddfnphfgcellkdfbfbjeloo
10 per POST aufgerufen werden und die folgende Variablen mit übergeben werden:
11 - token (Der Token des Gerätes, zum testen einfach einen String)
12 - id (die SplusName's der Vorlesungen, zum testen auch einfach einen oder mehrere Strings)
13
14 */
15
16
17 /* we can request debug output to better find errors */
18 $debug=0;
19 if ( isset( $_REQUEST['debug'] ))
20 {
21     $debug=1;
22     mysqli_report(MYSQLI_REPORT_ALL);
23
24     ini_set('mysql.trace_mode', 'On' );
25     ini_set('mysqli.trace_mode', 'On' );
26
27     ini_set('error_reporting', E_ALL | E_STRICT | E_DEPRECATED | E_NOTICE | E_PARSE );
28
29     ini_set('display_errors', 'On' );
30     ini_set('display_startup_errors', 'On' );
31
32     //ini_set('allow_call_time_pass_reference', 'On' );
33     ini_set('html_errors', 'On' );
34
35     ini_set('mysql.log_queries_not_using_indexes','on');
36     ini_set('mysql.log_slow_admin_statements','on');
37     ini_set('mysql.slow_query_log','on');
38     ini_set('mysql.log_error_verbosity','3');
39
40     ini_set('mysqli.log_queries_not_using_indexes','on');
41     ini_set('mysqli.log_slow_admin_statements','on');
42     ini_set('mysqli.slow_query_log','on');
43     ini_set('mysqli.log_error_verbosity','3');
44
45     // E_NOTICE ist sinnvoll um uninitialisierte oder
46     // falsch geschriebene Variablen zu entdecken
47     error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE | E_STRICT );
48
49 }
50
51

```

Anhang 2: fcm\_register\_user.php -Datei Teil 1

```

52 require_once "fcm_connect_db.php";
53
54 //BSP SQL-Injection
55 //$login = $this->mysqli->real_escape_string( $login );
56
57 // Alle übergebenen Parameter entwerten, um SQL-Injection aus dem Weg zu gehen
58 $fcm_token = htmlentities( $_POST["fcm_token"] );
59 // passiert hier weiter unten
60 $lectureJSON = $_POST["vorlesung_id"];
61 $lectureArray = json_decode($lectureJSON,true);
62
63
64 if ($debug) { echo "\nToken: $fcm_token\n\nlectureJSON: $lectureJSON\n";}
65
66 //Alle Einträge mit diesem Token in DB löschen
67 $sqldelete = "DELETE FROM fcm_nutzer WHERE token = \"\$fcm_token\" ";
68 $con->query($sqldelete);
69
70 //Tokens und Vorlesungen in DB eintragen
71 for ($i = 0; $i < count($lectureArray); $i++)
72 {
73     // htmlentities wegen SQL-Injection
74     // da htmlentities falsch encodet filter_var genommen
75     $vorlesung_id = filter_var($lectureArray[$i]['vorlesung_id'], FILTER_SANITIZE_STRING);
76     if ($debug) { echo "\nVorlesung_id: $vorlesung_id\n";}
77     $sqlinsert = "INSERT INTO `fcm_nutzer` (`token`, `vorlesung_id`) VALUES (\"\$fcm_token\",N'$vorlesung_id')";
78     $con->query($sqlinsert);
79 }
80
81 // SQLi-Connction schließen
82 $con->close();
83
84 return("Funktioniert!");
85 ?>

```

Anhang 3: fcm\_register\_user.php -Datei Teil 2

```

1  <?php
2
3  require_once "fcm_connect_db.php";
4  require_once "server.php";
5
6  $sql = "SELECT vorlesung_id FROM fcm_nutzer GROUP BY vorlesung_id";
7  $mySQLresult = $con->query($sql);
8  $vorlesung_ids = array();
9
10 //Vorlesungs_ids werden in vorlesung_ids Array gespeichert
11 echo"\n\nVorlesung ids: \n\n";
12 if($mySQLresult->num_rows >0) {
13     while($row = $mySQLresult->fetch_assoc()) {
14         echo "id: ";
15         echo $row['vorlesung_id'];
16         echo "\n";
17         array_push($vorlesung_ids, $row['vorlesung_id']);
18     }
19 }
20 $mySQLresult->close();
21
22 echo "Es wurden ".count($vorlesung_ids)." vorlesungs_ids durchsucht!\n";
23
24 //Es werden zu jeder vorlesung_id die verlegung_id's geholt
25 for ($i=0; $i < count($vorlesung_ids); $i++) {
26     $response = getChanges("", "", "", array($vorlesung_ids[$i]));
27     $countChanges = count($response['changes']);
28
29     echo "-----\n\n";
30
31     if($countChanges > 0){
32         echo "Für die vorlesungs_id ".$vorlesung_ids[$i]." \nliegen $countChanges Änderungen vor!\n";
33         //var_dump($response);
34     }
35     else{
36         echo "Für die vorlesungs_id ".$vorlesung_ids[$i]." \nliegen KEINE Änderungen vor!\n";
37     }
38
39     //Alle verlegung_id's werden ausgelesen
40     echo"\nverlegung_ids:\n\n";
41     for ($j = 0; $j < $countChanges; $j++) {
42         $verlegung_id = $response['changes'][$j]['splusname'];
43         echo "Id: ";
44         echo $verlegung_id;
45         echo "\n";
46
47         //Überprüfen, ob Pärchen aus vorlesung_id und verlegung_id in fcm_verlegungen schon vorkommt, falls nein -> reinschreiben
48         $sql2 = "SELECT * FROM fcm_verlegungen WHERE verlegung_id = '$verlegung_id' AND vorlesung_id = '$vorlesung_ids[$i]'";
49         $mySQLresult2 = $con->query($sql2);
50         if ($mySQLresult2->num_rows >0) {
51             echo "id wurde bereits eingetragen!\n";
52         }
53         else {
54             echo "id wird eingetragen!\n";
55             $sqlinsert = "INSERT INTO fcm_verlegungen (`verlegung_id`, `vorlesung_id`) VALUES ('$verlegung_id', '$vorlesung_ids[$i]'";
56             $con->query($sqlinsert);
57             //Sende Notifications an vorlesungs_id
58             sendNotification($vorlesung_ids[$i], $con, $response['changes'][$j]['label']);
59         }
60         $mySQLresult2->close();
61         echo "\n";
62     }
63 }

```

Anhang 4: fcm\_update\_and\_send.php -Datei Teil 1

```

64
65 // SQLi-Connction schließen
66 $con->close();
67
68 // Funktionen
69 // -----
70 function sendNotification($vorlesung_id, $con, $label) {
71     $sql3 = "SELECT token FROM fcm_nutzer WHERE vorlesung_id = '".$vorlesung_id."'";
72     $mySQLresult3 = $con->query($sql3);
73     $tokenArray = array("");
74
75     //Alle Tokens auslesen und in $tokens speichern
76     if ($mySQLresult3->num_rows > 0) {
77         //output data of each row
78         while ($row = $mySQLresult3->fetch_assoc()) {
79             if ($tokenArray[0] == '') {
80                 echo "Token leeres Feld: $row[token]\n";
81                 $tokenArray[0] = $row[token];
82             }
83             else {
84                 echo "Token hinzufügen: $row[token]\n";
85                 array_push($tokenArray, $row[token]);
86             }
87         }
88
89         //Nachricht senden mit jedem Token aufrufen
90         for($i=0; $i < count($tokenArray); $i++) {
91             sendGCM($tokenArray[$i], $label);
92             echo($tokenArray[$i]."<br>");
93         }
94         echo "Notification an Vorlesung_id $vorlesung_id wurde gesendet!<br>\n";
95     } else{
96         echo "Es sind keine Tokens für die vorlesungs_id <b>$vorlesung_id</b> vorhanden!<br>\n";
97     }
98     $mySQLresult3->close();
99 }
100

```

Anhang 5: fcm\_update\_and\_send.php -Datei Teil 2

```

100
101 //SendGoogleCloudMessage
102 function sendGCM($registration_ids, $label) {
103     //titel und message der Notification
104     $title = "Neue Änderung";
105     $message = "für das Fach ".$label;
106
107     //FCM URL
108     $url = "https://fcm.googleapis.com/fcm/send";
109     //$server_key = "AIzaSyAqg-02MqKHK4P9kWTfFaeX18AZuKz3-oH8";
110     $server_key = "AIzaSyCLUsCj9AxHYig3S_Gfu_ccC_9Y5Ii38zw";
111
112     //prepare data
113     $fields = array (
114         'registration_ids' => array($registration_ids),
115         'data' => array('notification_type' => 'change'),
116         'notification' => array('title' => $title, 'body' => $message, 'sound' => 'default')
117     );
118
119     $fields = json_encode ( $fields );
120
121     //header data
122     $headers = array ('Authorization: key='.$server_key, 'Content-Type: application/json');
123
124     //initiate curl request
125     $ch = curl_init();
126     curl_setopt($ch, CURLOPT_URL, $url);
127     curl_setopt($ch, CURLOPT_POST, true);
128     curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
129     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
130     curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
131
132     // execute curl request
133     $result = curl_exec($ch);
134
135     //close curl request
136     curl_close($ch);
137
138     echo "Token wurde an Google Firebase übermittelt: ";
139     //return output
140     return $result;
141 }
142 ?>

```

Anhang 6: fcm\_update\_and\_send.php -Datei Teil 3